Calculer des harmoniques sphériques
en **C++** avec **Boost::math::special_functions**
et en **Python** avec la librairie **Sympy**

Peter Philippe

29/05/2025

Introduction

Une façon imagée de définir les harmoniques sphériques, serait de dire qu'elles sont à la sphère ce que les sinus et cosinus sont au cercle, ou encore par analogie, que les harmoniques sphériques sont les sinus et cosinus dans $\mathbb{R}^3$ des séries de Fourier dans $\mathbb{R}^2$ d'une fonction de périodicité 2π.

Les applications des harmoniques sphériques sont nombreuses dans des domaines aussi pointus et variés que la physique quantique avec, par exemple, la célèbre équation de Schrödinger en dimension trois permettant d'étudier la position d'un électron par rapport à son noyau dans un atome d'hydrogène, en image de synthèse elles interviennent notamment dans la représentation de la fonction de distribution de la réflectance bidirectionnelle, en sismologie où leur rôle consiste, entre autres, à mettre en valeur l'écart tridimensionnel des zones d'élévations surfaciques d'une sphère en fonction des mouvements de l'écorce terrestre (topographie), etc.

Ce court document comprend trois sections :

- ✓ Les polynômes de Legendre associées.
- ✓ Les harmoniques sphériques.
- ✓ Sources en C++ et en Python avec Sympy et Matplotlib (pour la visualisation).

Tous les détails techniques seront passés sous silence, cela risquera d'agacer fortement les personnes expertes sur ce sujet ; qu'elles me pardonnent d'avoir pris cette liberté.

1 Les polynômes de Legendre associées

Ces polynômes P_n^m sont définis dans $[-1,1]$ par leur degré $n \in \mathbb{N}$ et un ordre $m \in \mathbb{Z}$ où $-n \leqslant |m| \leqslant n$ tel que $m = -n, -n+1, \ldots, n, n-1, n$, ils sont les solutions de l'équation différentielle

$$(1-x^2)y'' - 2xy' + y\left[n(n+1) - \frac{m^2}{1-x^2}\right] = 0 \quad n = 0,1,2,\ldots$$

La version du premier ordre est fréquemment représentée par une forme bien connue dite de Rodrigues :

$$\boxed{P_n^m(x) = \frac{(-1)^m}{2^n n!}(1-x^2)^{\frac{m}{2}} \frac{d^{n+m}}{dx^{n+m}}(x^2-1)^n} \tag{1}$$

Lorsque P_n^0, on retrouve les polynômes de Legendre P_n de degré n (utilisé notamment en intégration numérique avec la quadrature de Gauss), de même que :

- $P_0^0 = 1$ pour $n = m = 0$
- $P_n^m = 0$ pour $m > n$
- $P_m^m(x) = (-1)^m(2m-1)!!(1-x^2)^{\frac{m}{2}}$ pour $n = m$
- $P_{m+1}^m(x) = x(2m+1)P_m^m(x)$ pour $n = m+1$
- $P_n^n(x) = (2n)!(2^n n!)^{-1}(1-x^2)^{\frac{n}{2}}$ pour $m = n$
- $P_n^{m+1}(x) = 2mx(1-x^2)^{-1/2}P_n^m(x) + [n(n+1) - m(m-1)]P_n^{m-1}(x)$ pour $m+1$

$x!!$ correspond à la factorielle double.

Il est possible de rapprocher les fonctions de Legendre P_n^m avec les polynômes de Legendre P_n :

$$P_n^m(x) = (x^2-1)^{\frac{m}{2}} \frac{d^m}{dt^m} P_n(x)$$

Lorsque $m \geqslant -l$, la relation de proportionnalité entre $P_n^{-m}(x)$ et $P_n^m(x)$ est donnée par :

$$\boxed{P_n^{-m}(x) = (-1)^m \frac{(n-m)!}{(n+m)!} P_n^m(x)} \tag{2}$$

De même, lorsque x est négatif on a

$$P_n^m(-x) = (-1)^{n+m} P_n^m(x)$$

Ces fonctions sont normalisées $-1 \leqslant x \leqslant 1$, leur relation d'orthogonalité est $\forall\, n \geqslant m$:

$$\int_{-1}^{1} P_n^m(x) P_{n'}^m(x)\, dx = \begin{cases} 0 & \text{si } n \neq n' \\ \dfrac{2}{(2n+1)} \dfrac{(n+m)!}{(n-m)!} & \text{si } n = n' \end{cases} \tag{3}$$

En posant $x = \cos\theta$, les polynômes de Legendre peuvent se reformuler en terme trigonométriques :

$$P_n^m(\cos\theta) = (-1)^m (\sin(\theta))^m \frac{d^m}{d(\cos\theta)^m} \left(P_n(\cos\theta)\right)$$
$$P_n^{-m}(\cos\theta) = (-1)^m \frac{(n-m)!}{(n+m)!} P_n^m(\cos\theta)$$

Pour $n+m$ impair :

$$P_n^m(0) = 0$$

Pour $n+m$ pair :

$$P_n^m(0) = (-1)^{(n-m)/2} \frac{(n+m)!}{2^n [(n-m)/2]! [(n+m)/2]!}$$

L'une des relations de récurrence permettant de calculer les séries est :

$$(n-m) P_n^m(x) = x(2n-1) P_{n-1}^m(x) - (n+m-1) P_{n-2}^m(x)$$

Les premiers polynômes de Legendre associés sont pour $x = \cos\theta$:

$\mathbf{P_n^m(x)}$	
$P_0^0(x)$	1
$P_1^0(x)$	$\cos\theta$
$P_1^1(x)$	$-\sqrt{1-x^2} = -\sin\theta$
$P_2^0(x)$	$\frac{3x^2-1}{2} = \frac{3\cos^2\theta - 1}{2}$
$P_2^1(x)$	$-3x\sqrt{1-x^2} = -3\cos\theta\sin\theta$
$P_2^2(x)$	$3(1-x^2) = 3\sin^2\theta$
$P_3^0(x)$	$\frac{5x^3-3x}{2} = \frac{5\cos^3\theta - 3\cos\theta}{2}$
$P_3^1(x)$	$-\frac{3}{2}(5x^2-1)\sqrt{1-x^2} = -\frac{3}{2}(5\cos^2\theta - 1)\sin\theta$

$P_3^2(x)$	$15x(1-x^2) = 15\cos\theta\sin^2\theta$
$P_3^3(x)$	$-\sqrt[3]{15(1-x^2)} = -15\sin^3\theta$

2 Les harmoniques sphériques

Elles permettent de représenter une fonction sur la sphère. Une fonction sphérique se définit sur la sphère unité S^2 de $\mathbb{R}^3$ telle que

$$S^2 = \Big[(x,y,z) \in \mathbb{R}^3, \quad x^2+y^2+z^2 = 1\Big]$$

Les harmoniques sphériques sont des fonctions continues et invariantes par rotation, elles forment une base hilbertienne (orthonormale) pour la norme $\mathbf{L_2}$. Elles ont un lien de parenté très fort avec les polynômes orthogonaux. Les indices m et n représentant respectivement l'ordre et le degré, sont repris des polynômes de Legendre associées. Ces fonctions ont un laplacien nul.

Une fonction harmonique f est la solution de l'équation particulière de Laplace, ici, en coordonnées sphérique r,θ,ϕ :

$$\Delta f(r,\theta,\phi) = \nabla^2 f(r,\theta,\phi) = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\,\frac{\partial f}{\partial r}\right) + \frac{1}{r^2\,\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\,\frac{\partial f}{\partial\theta}\right) + \frac{1}{r^2\,\sin^2\theta}\frac{\partial^2 f}{\partial\phi^2} = 0$$

La décomposition d'une fonction harmonique f de classe $\mathbf{C}^2$ sur S^2 est représentée par la combinaison linéaire suivante :

$$\boxed{f(\theta,\phi) = \sum_{n=0}^{\infty}\sum_{m=-n}^{n} C_n^m Y_n^m(\theta,\phi)} \tag{4}$$

avec

$$C_n^m = \int_0^{2\pi}\int_0^{\pi} f(\theta,\phi) Y_n^m(\theta,\phi) d\theta d\phi$$

Les C_n^m sont des valeurs complexes et sont reliées entres elles par la relation

$$\mathbf{C}_n^m = (-1)^m\overline{C_n^m}$$

Et Y_n^m représente l'équation d'une harmonique sphérique :

$$Y_n^m(\theta,\phi) = \begin{cases} (-1)^m N_n^m P_n^m \cos\theta e^{i\phi} & \text{si} \quad 0 < m \leqslant n \\ N_n^0 P_n^0 \cos\theta & \text{si} \quad m = 0 \\ N_n^m P_n^{-m} \cos\theta e^{i\phi} & \text{si} \quad -n \leqslant m < 0 \end{cases}$$

où les angles θ et ϕ correspondent respectivement à la longitude et à la latitude.

Une harmonique sphérique Y_n^m résulte du produit de deux fonctions, la première N_n^m est dite de normalisation

$$\boxed{N_n^m = \sqrt{\frac{(2n+1)}{4\pi}\frac{(n-m)!}{(n+m)!}}} \tag{5}$$

Ces fonctions évitent ainsi de se retrouver à effectuer des calculs avec des coefficients dont les amplitudes respectives seraient beaucoup trop disparates, la seconde fonction est celle des polynômes de Legendre associées P_n^m vus lors de la section précédente :

$$P_n^m(x) = \frac{(-1)^m}{2^n n!}(1-x^2)^{\frac{m}{2}}\frac{d^{n+m}}{dx^{n+m}}(x^2-1)^n$$

De manière plus générale, une harmonique sphérique dans $\mathbb{R}^3$ se note :

$$\boxed{Y_n^m(\theta,\phi) = (-1)^m N_n^m P_n^m(\cos\theta)e^{i\phi}} \tag{6}$$

Pour $m = 0$, elle se simplifie :

$$Y_n^0(\theta,\phi) = \sqrt{\frac{2n+1}{4\pi}}P_n(\cos(\theta))$$

Lorsque m est négatif, les fonctions Y_n^m sont liées par une relation de symétrie :

$$Y_n^{-m}(\theta,\phi) = (-1)^m(Y_n^m)^*(\theta,\phi)$$

Ainsi que par une relation de parité pour les directions opposées :

$$Y_n^{-m}(\pi-\theta,\phi+\pi) = (-1)^n(Y_n^m)(\theta,\phi)$$

Le symbole * représente le conjugué au sens des nombres complexes.

$\mathbf{C}_n^m$ est souvent appelé coefficient de Fourier (éventuellement *normalisé*) et les P_n^m sont les polynômes de Legendre associés. Ces fonctions de degré n et d'ordre m sont orthonormales sur la sphère et leurs valeurs sont alternativement positives et négatives.

Un cas spécial est

$$Y_n^m(\theta,\phi) = \begin{cases} \left(\dfrac{2n+1}{4\pi}\right)^{1/2} & \text{si} \quad m = 0 \\ 0 & \text{si} \quad m = 1,2,3,\ldots \end{cases}$$

Voici quelques exemples de développements des premières harmoniques sphériques $Y_n^m(\theta,\phi)$:

$\mathbf{Y_l^m(x)}$	**Développement**
$Y_0^0(x)$	$\frac{1}{\sqrt{4\pi}}$
$Y_1^0(x)$	$\sqrt{\frac{3}{4\pi}}\cos\theta$
$Y_1^{\pm1}(x)$	$\mp\sqrt{\frac{3}{8\pi}}\sin\theta\ e^{\pm i\phi}$
$Y_2^0(x)$	$\frac{1}{2}\sqrt{\frac{5}{4\pi}}(3\cos^2(\theta)-1)$
$Y_2^{\pm1}(x)$	$\mp\sqrt{\frac{15}{8\pi}}(\sin\theta\ \cos\theta\ e^{\pm i\phi})$
$Y_2^{\pm2}(x)$	$\frac{1}{4}\sqrt{\frac{15}{2\pi}}(\sin^2\theta\ e^{\pm 2i\phi})$
$Y_3^0(x)$	$\frac{1}{4}\sqrt{\frac{7}{\pi}}(5\cos^3\theta-3\cos\theta)$
$Y_3^{\pm1}(x)$	$\mp\frac{1}{2}\sqrt{\frac{21}{4\pi}}\sin\theta\ (5\cos^2\theta-1)\ e^{\pm 2i\phi}$
$Y_3^{\pm2}(x)$	$\frac{1}{2}\sqrt{\frac{105}{8\pi}}\sin^2\theta\ \cos\theta\ e^{\pm 2i\phi}$
$Y_3^{\pm3}(x)$	$\mp\frac{1}{2}\sqrt{\frac{35}{4\pi}}\sin^3\theta\ e^{\pm 3i\phi}$

Plusieurs centaines de tracés d'harmoniques sphériques sont visibles à :
www.lifesmith.com/spharmin.html.

Selon que $Y_n^m(\theta,\phi)$ ou $|Y_n^m(\theta,\phi)|$ l'harmonique sera symétrique ou non par rapport à $m=0$.

Dans certains cas, il est nécessaire de changer une fonction de position (comme un changement d'axe par exemple). On effectue alors une rotation sur la sphère de chaque coefficient sphérique, matriciellement cela donne :

$$\begin{pmatrix} Y_n^m(\theta_0,\phi_0+\alpha) \\ Y_n^{-m}(\theta_0,\phi_0+\alpha) \end{pmatrix} = \begin{pmatrix} \cos m\alpha & -\sin m\alpha \\ \sin m\alpha & \cos m\alpha \end{pmatrix} \begin{pmatrix} Y_l^m(\theta_0,\phi_0) \\ Y_l^{-m}(\theta_0,\phi_0) \end{pmatrix}$$

3 Implémentation en C++ avec Boost

Ce code très simple ne permet d'effectuer que des calculs selon les valeurs de n, m, θ et ϕ :

```
/*
# Fichier Makefile
CXX      := g++-14
CXXFLAGS := -march=native -flto=auto -O3 -std=gnu++23 -Wall
INCFLAGS := -I/usr/local/include/
LDFLAGS  := -L/usr/local/lib/
LDLIBS   := -lmpfr

TARGET   := harm_sph
SOURCE   := harm_sph.cpp

all: $(TARGET)

$(TARGET): $(SOURCE)
  $(CXX) $(CXXFLAGS) $(INCFLAGS) $(LDFLAGS) $(LDLIBS) $(SOURCE) -o $(TARGET)

clean:
  rm -f $(TARGET)
*/

#include <cstdlib>
#include <complex>
#include <cmath>
#include <limits>
#include <print>
#include <boost/math/special_functions/spherical_harmonic.hpp>

int main(int argc, char* argv[]) {
    if (argc != 5) {
        std::println(stderr,
            "Usage : {} <n> <m> <theta> <phi>\n"
            "       where: n ⩾ 0, |m| ⩽ n, theta ∈ [0;π ], phi ∈ [0;2π ] (en radians)",
            argv[0]
        );
        return EXIT_FAILURE;
    }

    int n = std::atoi(argv[1]);
    int m = std::atoi(argv[2]);
    float theta = std::strtof(argv[3], nullptr);
    float phi  = std::strtof(argv[4], nullptr);

    if (n < 0) {
        std::println(stderr, "Erreur : n doit être ⩾ 0");
        return EXIT_FAILURE;
    }
```

```
    if (std::abs(m) > n) {
        std::println(stderr, "Erreur : |m| doit être ⩽ n");
        return EXIT_FAILURE;
    }

    if (theta < 0.0f || theta > static_cast<float>(M_PI) ||
        phi < 0.0f || phi > static_cast<float>(2 * M_PI)) {
        std::println(stderr, "Erreur : theta doit être dans [0, π ] et phi dans [0, 2π ]");
        return EXIT_FAILURE;
    }

    std::complex<float> Y = boost::math::spherical_harmonic<float>(n, m, theta, phi);

    std::println("Y_{}^{}(theta={:.6f}, phi={:.6f})", n, m, theta, phi);
    std::println("Partie réelle : {:.6f}", Y.real());
    std::println("Partie imaginaire : {:.6f}", Y.imag());

    return EXIT_SUCCESS;
}
```

Voici deux exemples :

```
% ./harm_sph 3 -3 0.707 0.707
Y_3^-3(theta=0.707000, phi=0.707000)
Partie réelle : -0.059787
Partie imaginaire : -0.097470
% ./harm_sph 6 -5 0.866 0.5
Y_6^-5(theta=0.866000, phi=0.500000)
Partie réelle : -0.222772
Partie imaginaire : -0.166416
```

4 Implémentation en Python avec Sympy et Matplotlib

Ce petit programme en python, bien qu'assez limité, est un peu plus élaboré (même s'il ne propose plus de calculer les valeurs complexes), car il permet de visualiser les harmoniques sphériques selon n et m et permet d'afficher la version standard, la version réelle ou la version imaginaire :

```
from sys import argv
import numpy as np
import matplotlib.pyplot as mpl
from sympy import symbols, re, im, lambdify
from sympy.functions.special.spherical_harmonics import Ynm


def harmonique_spherique(n: int, m: int, c: str) -> None:
    symbole_theta = symbols("theta", real=True)
```

```
symbole_phi = symbols("phi", real=True)
y_lm = Ynm(n, m, symbole_theta, symbole_phi).expand(func=True)
y_re, y_im = re(y_lm), im(y_lm)

resolution = 200
grille_theta = np.linspace(0, np.pi, resolution)
grille_phi = np.linspace(0, 2 * np.pi, resolution)
theta, phi = np.meshgrid(grille_theta, grille_phi)
magnitude = y_re**2 + y_im**2

if c == "":
    fonction_lambda = lambdify((symbole_theta, symbole_phi), magnitude, modules=["numpy"])
elif c == "r":
    fonction_lambda = lambdify((symbole_theta, symbole_phi), y_re, modules=["numpy"])
elif c == "i":
    fonction_lambda = lambdify((symbole_theta, symbole_phi), y_im, modules=["numpy"])
else:
    raise ValueError("Seules les lettres 'r' et 'i' sont autorisées !")

rayon = fonction_lambda(theta, phi)
rayon = np.where(rayon < 0, 0, rayon)
rho = rayon if c == "" else np.sqrt(rayon)

x = rho * np.sin(theta) * np.cos(phi)
y = rho * np.sin(theta) * np.sin(phi)
z = rho * np.cos(theta)

ax = mpl.figure(figsize=(8, 8)).add_subplot(111, projection="3d")
rayon_norme = np.interp(rayon, (rayon.min(), rayon.max()), (0.1, 0.9))
couleurs = mpl.cm.rainbow(rayon_norme)
ax.plot_surface(x, y, z, facecolors=couleurs, rstride=2, cstride=1, lw=1, antialiased=True)

min_max = np.array([[x.min(), x.max()], [y.min(), y.max()], [z.min(), z.max()]])
centres = np.mean(min_max, axis=1)
ext = (np.diff(min_max, axis=1) * 0.5).max()
ax.set_xlim(centres[0] - ext, centres[0] + ext)
ax.set_ylim(centres[1] - ext, centres[1] + ext)
ax.set_zlim(centres[2] - ext, centres[2] + ext)
ax.set_box_aspect([1,1,1])
ax.set_axis_off()

mode = ""
match (c):
    case "r":
        mode = " (réelle)"
    case "i":
        mode = " (imaginaire)"

mpl.title(f"Harmonique sphérique $Y_{{{n}}}^{{{m}}}(\\theta, \\phi)${mode}")
mpl.tight_layout()
# mpl.savefig(f"hs_{n}_{m}{mode}.jpg")
```

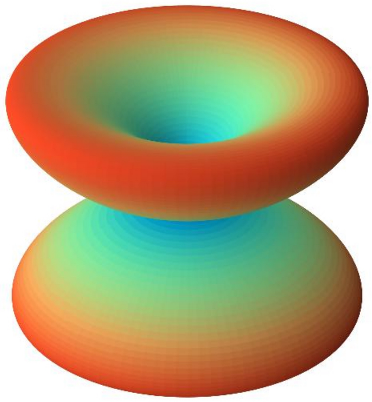
```
    mpl.show()

if __name__ == "__main__":
    len_args = len(argv)
    c = ""
    if not (3 <= len_args <= 4):
        print(f"Usage : python {argv[0]} l m 'r ou i'")
        exit(1)

    try:
        n = int(argv[1])
        m = int(argv[2])
        if len_args == 4:
            c = str(argv[3])
    except ValueError:
        print("Error: n and m doivent être des entiers.")
        exit(1)

    if np.fabs(m) > n:
        print(f"Erreur : |m| doit être ⩽ n : m = {m}, n = {n}")
        exit(1)

    harmonique_spherique(n, m, c)
```

Les six commandes suivantes

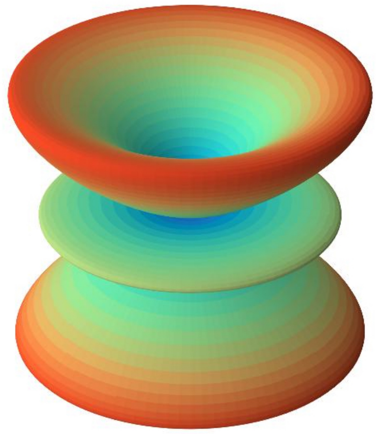
```
% python3 harm_sph.py 3 2
% python3 harm_sph.py 6 4
% python3 harm_sph.py 4 -1 r
% python3 harm_sph.py 5 -2 i
% python3 harm_sph.py 6 -3 r
% python3 harm_sph.py 7 -4 i
```

produisent les images ci-dessous :

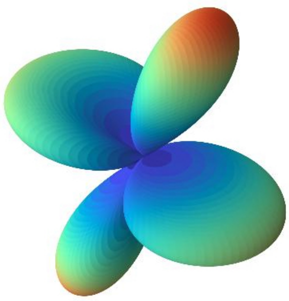
Harmonique sphérique $Y_3^2(\theta, \phi)$



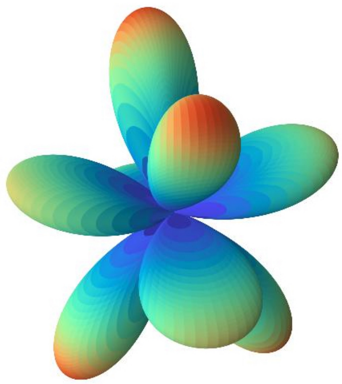
Harmonique sphérique $Y_6^4(\theta, \phi)$



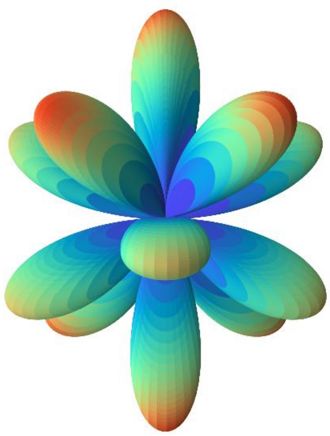
Harmonique sphérique $Y_4^{-1}(\theta, \phi)$ (réelle)



Harmonique sphérique $Y_5^{-2}(\theta, \phi)$ (imaginaire)



Harmonique sphérique $Y_6^{-3}(\theta, \phi)$ (réelle)



Harmonique sphérique $Y_7^{-4}(\theta, \phi)$ (imaginaire)